

# ナップザック問題に対する効率的 GA の適用

平林 永行・Michael Giovenco<sup>1</sup>・片山 謙吾<sup>2</sup>・成久 洋之<sup>3</sup>

岡山理科大学大学院工学研究科修士課程情報工学専攻

<sup>1</sup>Harvard University

<sup>2</sup>岡山理科大学大学院工学研究科博士課程システム科学専攻

<sup>3</sup>岡山理科大学工学部情報工学科

(1996年10月7日 受理)

## 1. 序 論

近年、生物の進化過程からヒントを得た遺伝的アルゴリズム (*Genetic Algorithm : GA*) が注目をあびている。GA は生物進化現象に潜む情報多様化、環境適応、様々な戦略などにみられる問題解決能力などの進化プロセスを、情報処理モデル化して自然の問題解決機構を理解し、利用したものである。GA は様々な問題に適用され実用的にも成果をあげている。GA をある問題に適用する場合に様々な問題点がある。その内の一つに、GA で要求される遺伝的オペレータなどの確率パラメータ設定がある。GA を効率よくその問題に適用するためには、それらの適切なパラメータ設定値が非常に大きな問題となる。そこで本研究では、組合わせ最適化問題であるナップザック問題を取り上げ、最も適切なパラメータ設定を検討するものである。

## 2. ナップザック問題

ナップザック問題とは、複数個の荷物の中からいくつかを、一定の容積を持った袋（ナップザック）に任意に選択して詰め込んでいったとき、ナップザックの容積内で最大限に詰め込める荷物の組合わせを求めるというものである。それは一般的な組合わせをすべて計算してから、その最大なものを決定するなどの方法では、膨大な時間が掛かるように設定されている。また、問題のサイズが大きくなれば、組合わせの爆発が生じる NP 完全なクラスに属する組合わせ最適化問題と考えられている。

## 3. 遺伝的アルゴリズム

遺伝的アルゴリズム (GA) は、生物進化（選択淘汰・突然変異）の原理に着想を得たアルゴリズムであり、確率的探査・学習・最適化の一手法と考えることができる。GA を最適化問題に適用する場合、各解候補は、染色体と呼ばれる有限長文字列に符号化しなければならない。染色体を構成する名文字列を遺伝子、染色体の集合を集団、集団に属する染

染色体数を集団サイズという。GA の特徴は、個体と呼ばれる複数の解候補を持ち、その各個体に対して最適関数、スケーリングを用いて、「交叉」、「突然変異」、「選択」、と呼ばれる遺伝的オペレータを、各世代ごとにその個体の持つ染色体や遺伝子にそれらの操作を確率的に行う、その生存競争に基づく確率的探索原理により、組合わせ問題のような計算量の多い問題の近似解または最適解を比較的容易に短時間で見つけることができるとされている。

#### 4. GA によるナップザック問題の実装

本実験で実装される GA では以下のような「コード化」、「適応度関数」、「遺伝的オペレータ」を採用した。

##### 4.1 処理手順

GA による処理手順は、以下のようになる。

1. 初期集団の生成
  - (a) コード化
2. 終了条件が満たされるまでループ
  - (a) 適応度の評価
  - (b) スケーリング
  - (c) 選択
  - (d) 交叉
  - (e) 突然変異

##### 4.2 コード化

ナップザック問題を解くにあたって、この問題の個体を文字列（染色体）にマッピング（コード化）しなければならない。マッピングはその物体の数のビット長を要求する。物体  $X_i$  を文字列の  $i$  番号のビット（遺伝子）に対応され、物体  $X_i$  が選択されている時を 1 で、そうでない時を 0 で表現する。例えば、物体の数が全部で 10 個ある場合 10 個のビットが準備され、5, 7 番目の物体が選択された状態は

“0 0 0 0 1 0 1 0 0 0”

と表すことが可能である。

##### 4.3 適応度関数

適応度関数は、個々の個体の競争力を表し、適応度関数を用いて宣言的に問題を記述で

きる。本研究では、ある解候補において選択した物体の価値の合計をそのまま適応度とするような関数を定義する。ただし、ナップザックにいれる物体の重さには上限があるので、上限を越すような選択に対してはペナルティーとして非常に悪い適応度を与える。これにより、意味のない解が生成されることを防ぐことができる。

(適応度関数)

$$f(\text{個体}) = \text{価値の合計} \quad \text{if } (\sum \text{選択した物の重さ} \leq \text{重さの上限})$$

$$= \text{十分悪い値} \quad \text{if } (\sum \text{選択した物の重さ} > \text{重さの上限})$$

#### 4.3.1 スケーリング (scaling)

スケーリングとは、適応度が決定されたとして、この値をそのまま選択時の確率に反映させる必要はなく、何らかの関数を導入し適応度の違いを拡大または縮小することである。本研究では、線形スケーリングを使用し適応度を表す。もとの適応度を  $f$ 、新たな適応度を  $f'$  とし以下の方程式

$$f' = af + b$$

で求める。

### 4.4 遺伝的オペレータ

#### 4.4.1 選択交配 (selection)

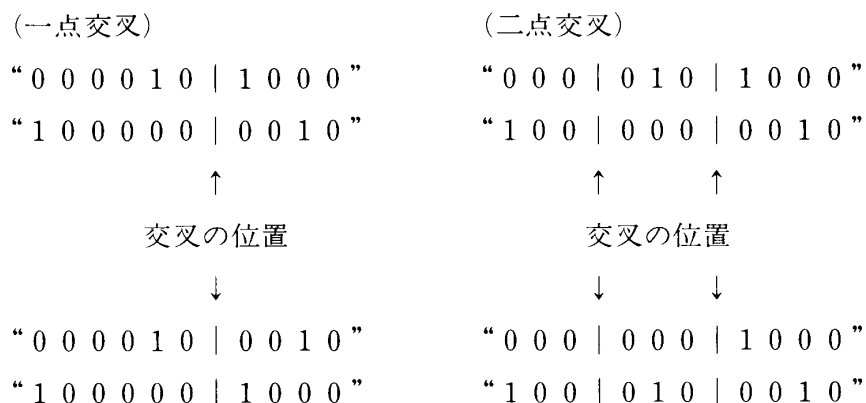
選択淘汰の圧力を与える際に重要なのが、どの個体同士を交配させるかであり、選択交配は、適応に応じてより環境に適した個体を確率的に選ぶ操作であるが、選択は、あくまでも確率的に行い、適応度の低いものであっても生き残れる可能性を残している。本研究では、適応度比例戦略を使用した。適応度比例戦略とは、各個体の適応度に比例した確率で子孫を残せる可能性を持っている。ある個体  $i$  が、各々の選択で選ばれる確率  $p_{select}$  は、以下の式

$$p_{selecti} = f_i / \sum_{j=1}^n f_j$$

で求められる。

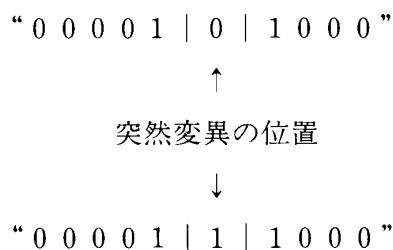
#### 4.4.2 交叉 (crossover)

交叉は、二つの親の染色体を組み替えて子の染色体を作る操作であり、二つの個体の間で文字列の部分的な交換を確率的に行う操作である。異なった文字列同士の交叉であれば広域的探索となり、似通った文字列同士の交叉であれば局所的探索となる。本研究では、交叉する位置を一つ決めてその前と後で、入れ替える一点交叉と、文字列をランダムに二ヶ所で切断し、中央部を入れ替える二点交叉を使用した。



#### 4.4.3 突然変異 (mutation)

突然変異は、遺伝子を一定の確率で変化させる操作であり、個体中の特定ビットを別のビットに確率的に変更する操作である。突然変異の役割は、交叉で得られた解の近傍を探索する操作と、交叉だけでは得ることのできないパターンを生成する操作である。本研究では、文字列の1ヶ所をランダムに選択し反転させる一点突然変異を使用した。



## 5. 実験結果

本研究で検討するパラメータ群は、最適な集団数、交叉・突然変異の確率と、一点交叉と二点交叉の場合の収束傾向の違いであり統計的なデータ収集を目的とした。扱うナップザック問題のサイズは50個に設定し、計算打ち切り世代数を500世代、集団数を30個から30個刻みで1000個までの34種類、交叉の確率を0.1から0.05刻みで1.0までの19種類、突然変異の確率を0.00001～0.1まで20種類の変化をさせた。この問題の最適解は356である。使用した計算機は *PC-9821 Xt13*、使用言語は *C* 言語である。

図1～10は本研究で得られた結果の一部で、集団数が一点交叉で30, 360, 720, 1000個の場合、二点交叉では30, 120, 360, 510, 720, 1000個の場合の交叉・突然変異の確率に対する収束傾向の変化である。図1～10の(a), (b), (c)の内容は

- (a) 交叉・突然変異の確率に対する適応度の関係
- (b) 交叉・突然変異の確率に対する世代の関係
- (c) 交叉・突然変異の確率に対する計算時間の関係

を示したものである。

## 6. 考 察

各結果図より集団数を増やすことによって最適解が確実に出力される。それによって全体的に計算時間および世代数を費やす傾向にある。しかしながらその計算によした時間や世代数が極端に短い確率ゾーンがある。それは、計算時間では突然変異の確率が0.07~0.005の間で短く、世代数では突然変異の確率が0.05~0.005の間で小さい。しかしそれは、集団数が500個を越えた辺りから現れる傾向にある。この確率ゾーンではほぼ100%の高い確率で最適解の算出を示してあり、効率的な探索がなされたと解釈できる。以上の傾向が算出し始めたのは、集団数が510個の場合からである。

集団数が少ない場合では、集団数が100個を越えた辺りから突然変異の確率が0.05~0.005の間で最適解が計算され始める。それ以外の突然変異の確率の場合では計算時間も短く世代数も小さいが最適解を得ることができない。最適解が算出される場合は、計算時間は短い、世代数が大きいという傾向がある。最初に最適解が算出された集団数は120個の場合である。

図11は本研究で得られた結果で、二点交叉を行い集団数だけを変化させ、その時の最大適応度と、それに要した計算時間の変化である。図11の内容は、

(a) 個体数に対する適応度と計算時間の関係  
を示したものである。

図12, 13は本研究で得られた結果で、二点交叉での集団数が120, 510個の場合の世代ごとの最大, 最小, 平均適応度の収束傾向である。図12, 13の内容は、

(a) 世代と適応度の関係  
を示したものである。

図11は、ほとんどの集団の場合に最適解が算出され、計算時間も多少ばらつきがあるがCPU-Time が 0 sec ~ 8 sec の間で最適解が算出されている。

図12, 13を比較してみると集団数が120個の時より、集団数が510個の方が速い世代で最適解に収束し、平均も全体的に最適解に近づいている。

## 7. 結 論

本研究は GA をナップザック問題に適用し、一点交叉・二点交叉について、集団数の変化、交叉・突然変異の確率の変化を各パラメータ群に対して最大適応度、計算時間などのデータを示した。その結果より、突然変異の確率では0.05~0.005の間で、集団数を増やしていくと速い計算時間で最適解が算出され、最初に算出される最適解の世代も速く算出されることがわかったが、突然変異の確率が0.005を越えると、突然変異の確率の影響がなくなってくることもわかった。交叉の確率の変化に関しては確率が増えることによって徐々に計算時間も最初に算出される最適解の世代も速く算出されることがわかった。以上のことより適応度、計算時間を考慮すると、本研究における推奨値は、二点交叉を用いて、集団

数は150個前後に取り，交叉の確率は0.6以上，突然変異の確率は0.05～0.005付近である。

なお本研究では，固定された問題サイズにおいて一点交叉・二点交叉についてのみ実験を行ったが，他の交叉方法，また問題サイズの大きさによっても同様の結果が得られるかどうか今後検討する予定である。

【一点交叉の場合の収束傾向】

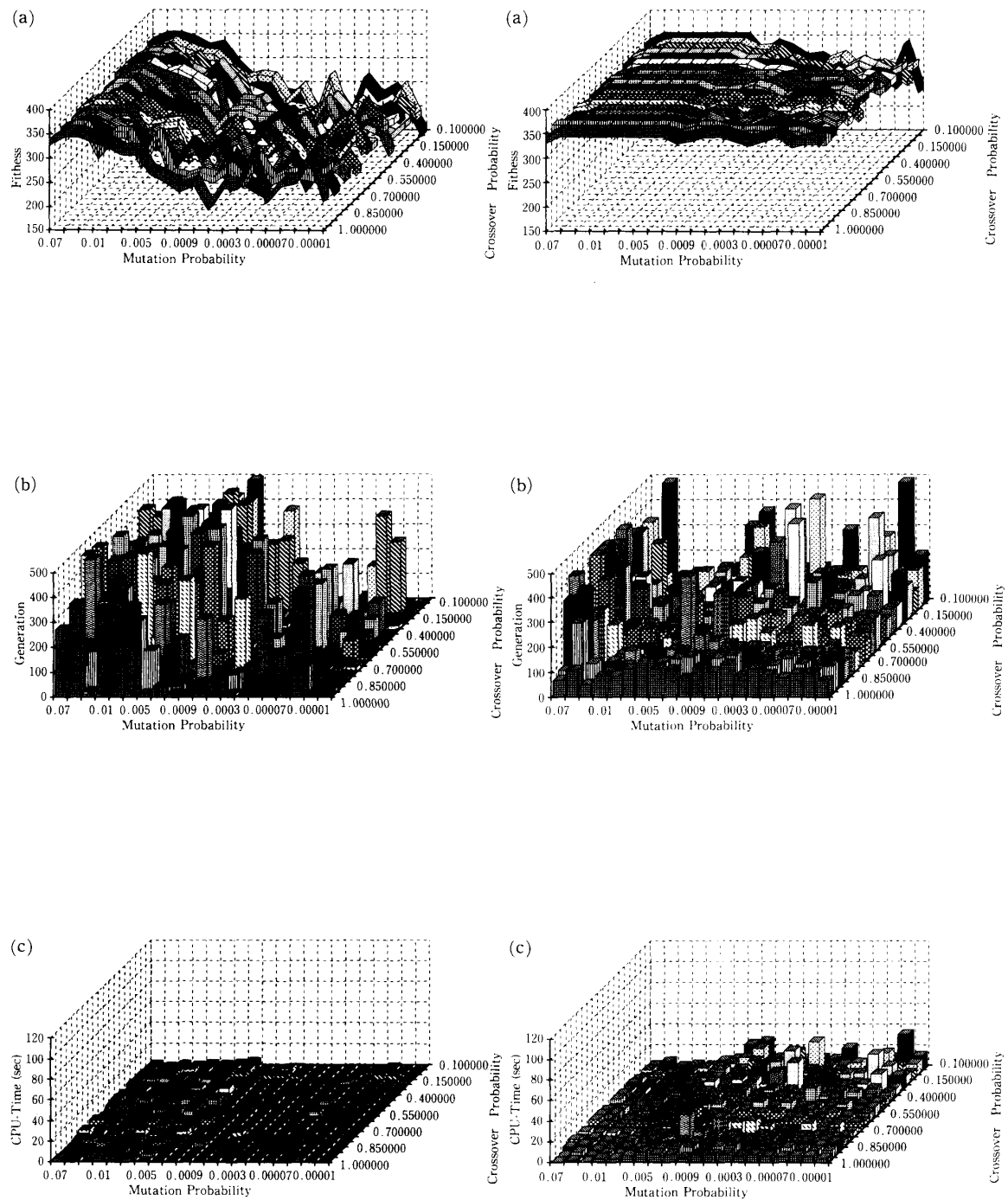


図 1 集関数が30個の時の収束傾向

図 2 集関数が360個の時の収束傾向

## 【一点交叉の場合の収束傾向】

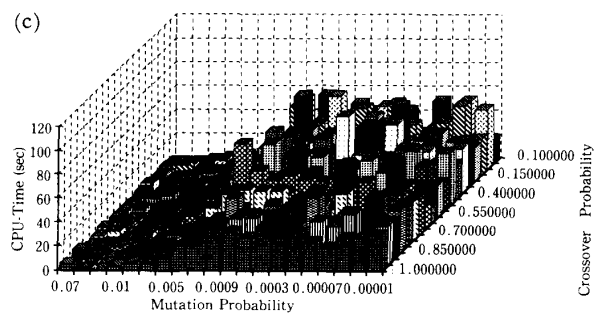
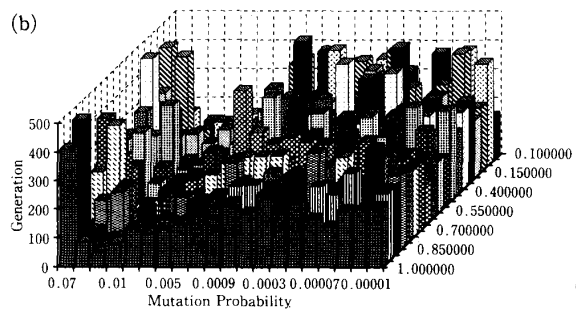
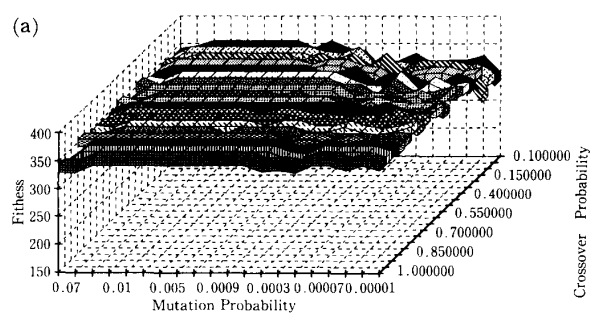


図3 集図数が720個の時の収束値向

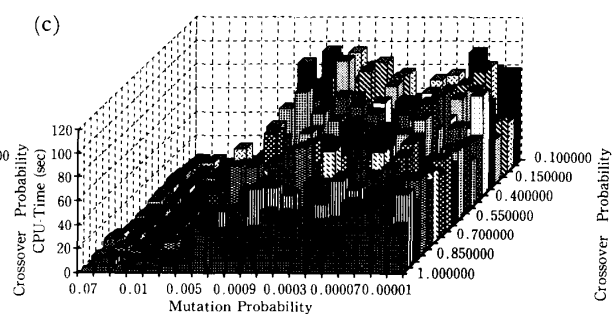
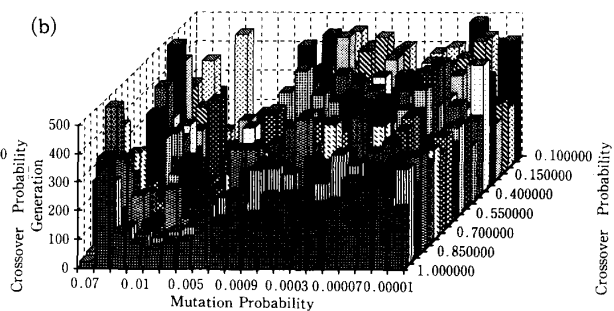
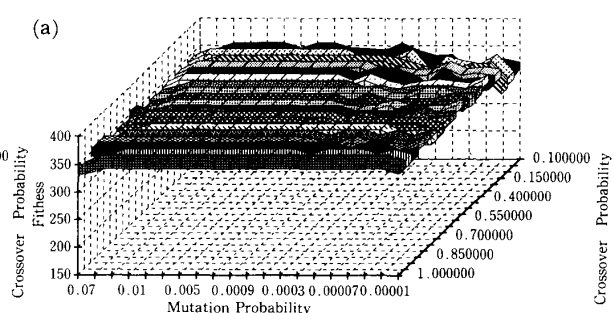


図4 集図数が1000個の時の収束値向



【二点交叉の場合の収束傾向】

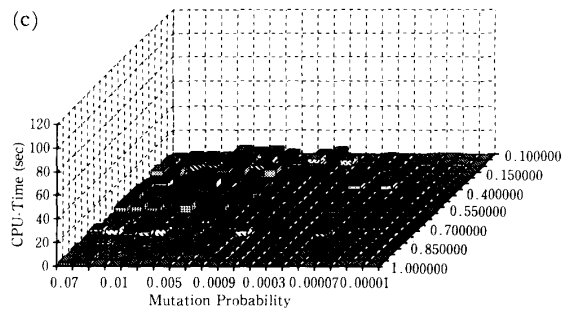
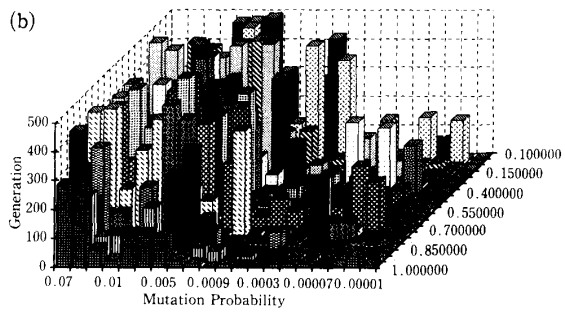
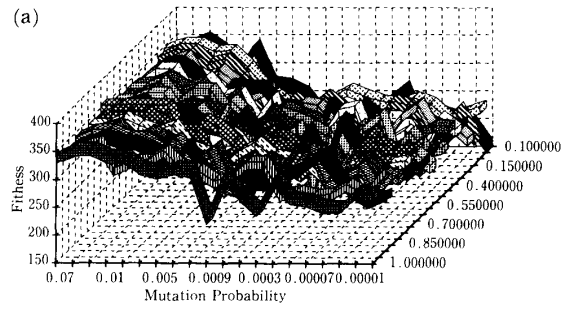


図 5 集関数が30個の時の収束値向

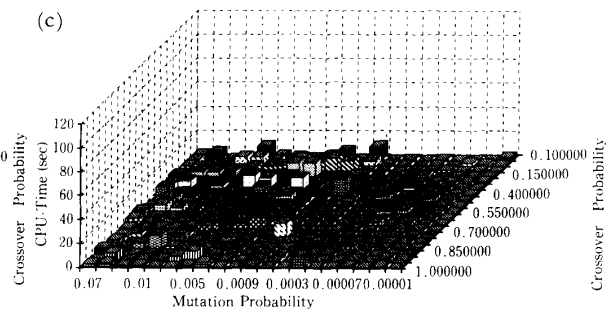
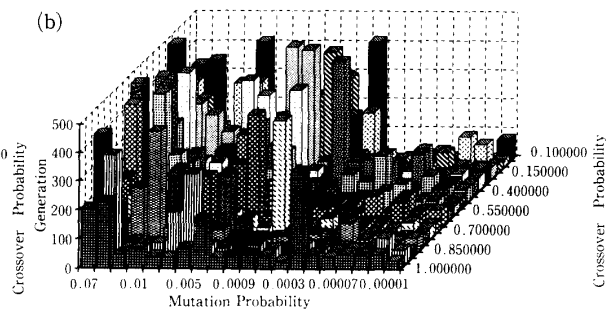
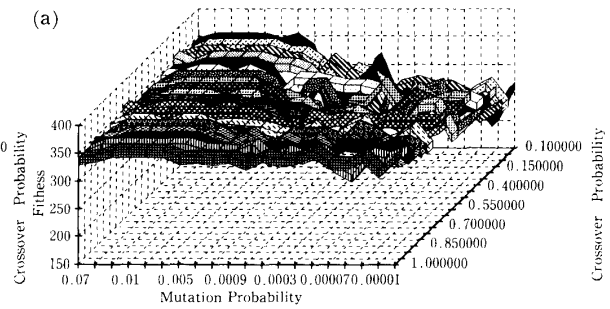


図 6 集関数が120個の時の収束値向

## 【二点交叉の場合の収束傾向】

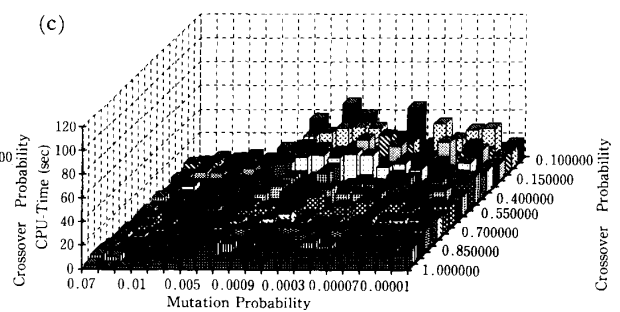
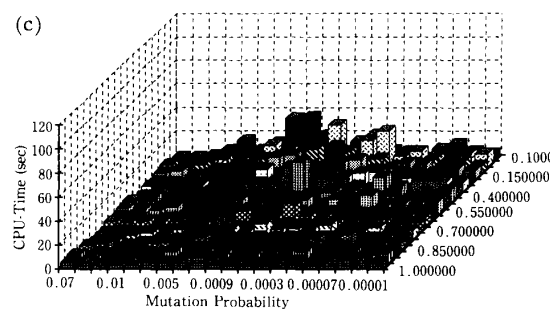
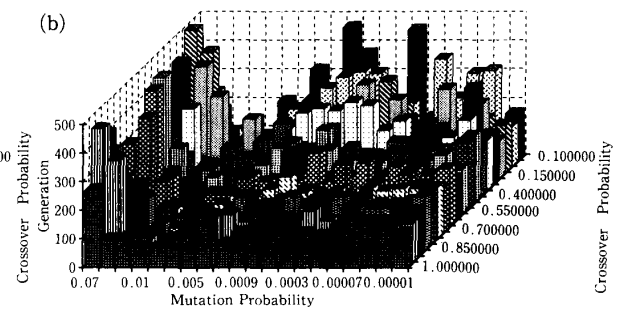
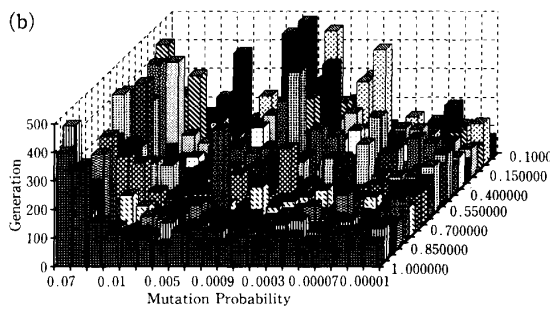
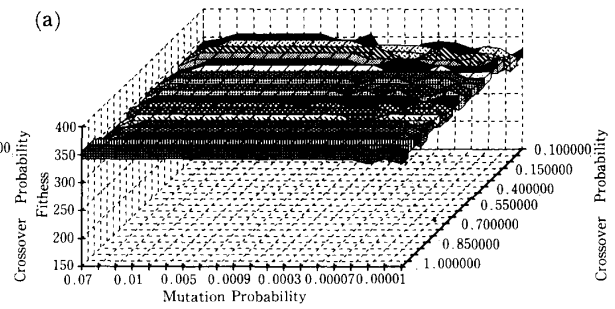
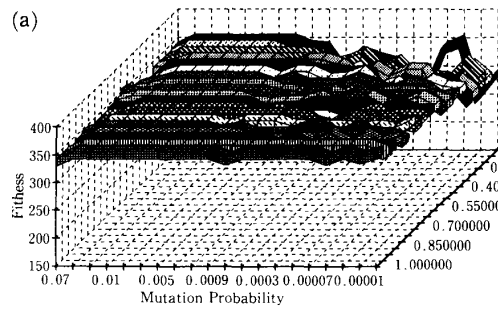


図 7 集関数が360個の時の収束値向

図 8 集関数が510個の時の収束値向

【二点交叉の場合の収束傾向】

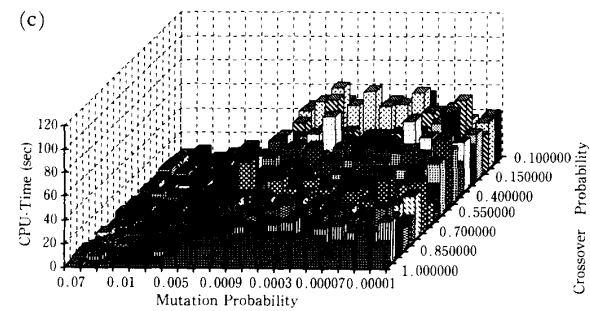
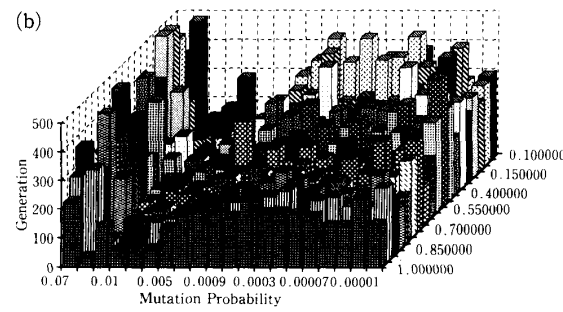
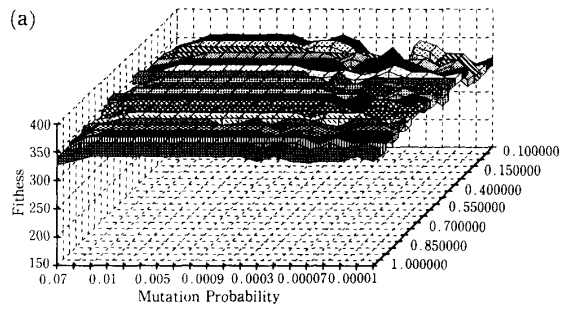


図9 集関数が720個の時の収束値向

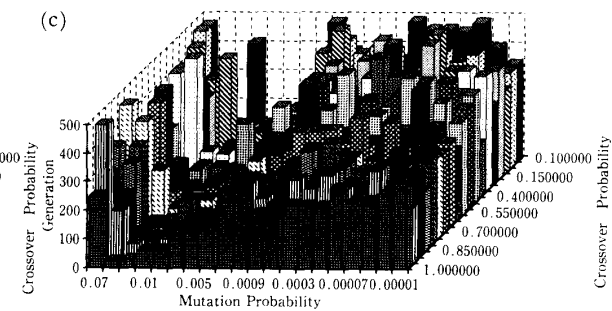
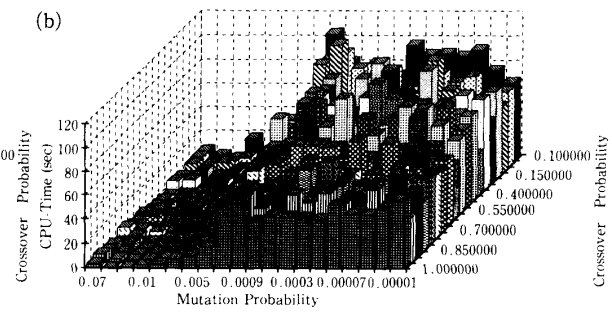
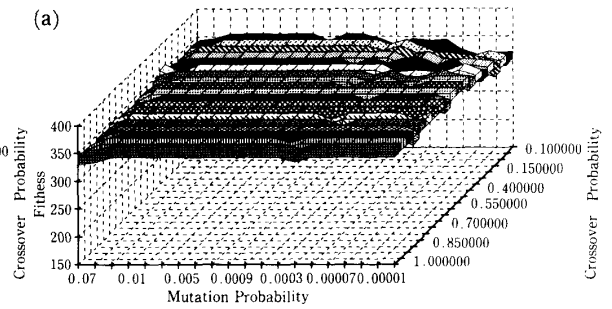


図10 集関数が1000個の時の収束値向

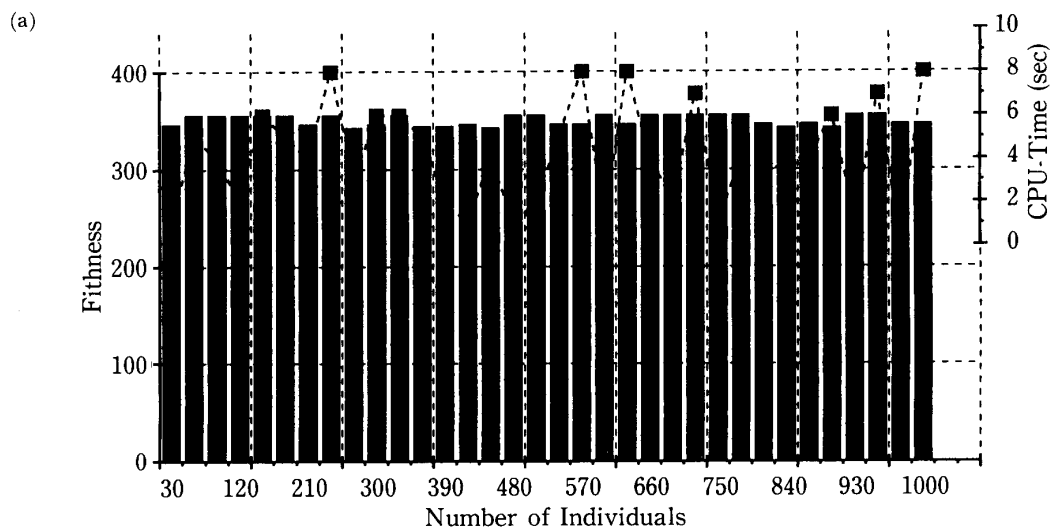


図11 交叉の確率が0.6、突然変異の確率が0.05の時の収束傾向

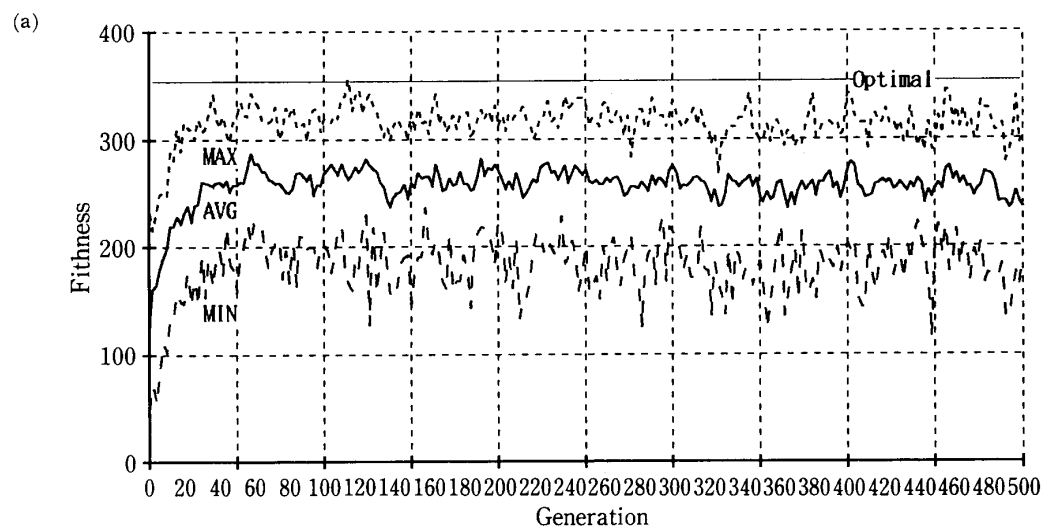


図12 集団数が120個の時の収束傾向

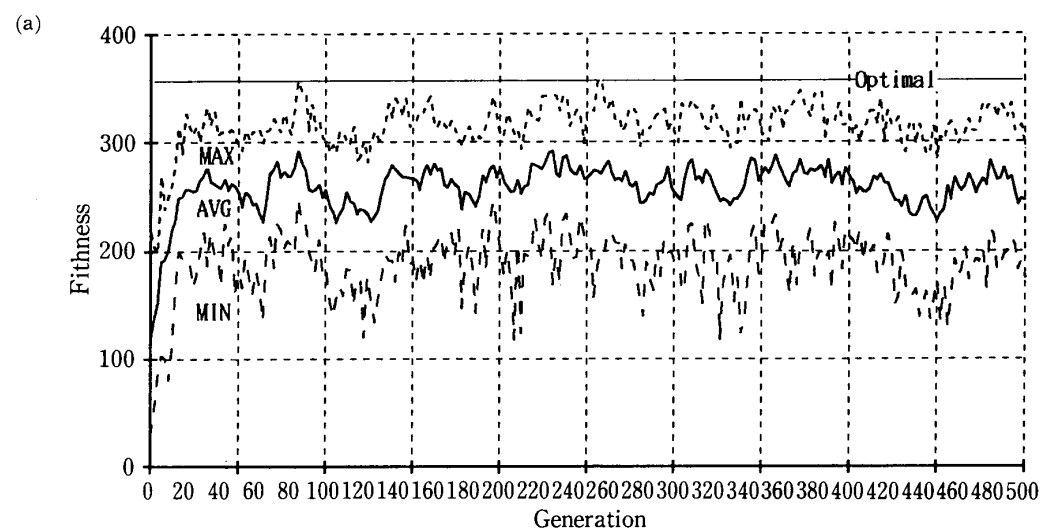


図13 集団数が510個の時の収束傾向

## Some Comments on Efficient Genetic Algorithms for Solution of Knapsack Problems

Hisayuki HIRABAYASHI, Michael GIOVINCO<sup>1</sup>,  
Kengo KATAYAMA and Hiroyuki NARIHISA<sup>2</sup>

*Graduate School of Engineering*

*<sup>1</sup>Harverd University*

*<sup>2</sup>Department of Information & Computer Engineering*

*Okayama University of Science,*

*Ridai-cho 1-1, Okayama 700, Japan*

(Received October 7, 1996)

Recently, Genetic Algorithms have been focussed a lot of attentions in search, optimization and machine learning as kind of artificial intelligence as well as neural networks algorithm.

Genetic Algorithms are search algorithms based on the mechanics of natural selection and natural genetics.

In practice, we must set up parameters of genetic operations, such as selection, crossover and mutation in order to obtain the final solution in applying genetic algorithms for a given optimization problem.

In this paper, we propose an efficient parameters of genetic operations for the given knapsack problems.